5

HIGH-QUALITY GRADIENT-CORRECTED LINEAR INTERPOLATION FOR DEMOSAICING OF COLOR IMAGES

10

by

Henrique S. Malvar

Li-wei He

15

and

Ross Cutler

20

TECHNICAL FIELD

The present invention relates in general to processing of digital color images and more particularly to method and system of linearly combining a

25   correction term with an interpolation to obtain a high-quality estimate of a missing pixel color at a pixel within the image.

BACKGROUND OF THE INVENTION

Digital cameras continue to steadily increase in popularity with consumers.

30   One reason is that the cost of digital cameras continues to decline, making them well within the reach of the average consumer, and causing more consumers to forego traditional film photography in favor of digital cameras. In order to contain cost, many consumer-grade digital color cameras are single-sensor digital cameras. As the name implies, in a single-sensor digital camera only a single

35   image sensor is used to capture color information for each pixel in a color image. Each image sensor, which is typically a charge-coupled device (CCD) or a

complementary metal oxide semiconductor (CMOS), is part of a sensor array that together represent the pixels of a color image.

Each image sensor can only generate information about a single color at a
5   given pixel. A color image, however, is represented by combining three separate monochromatic images. In order to display a color image, all of the red, blue and green (RGB) color values are needed at each pixel. In an ideal (and expensive) camera system, each pixel in the sensor array would be provided with three image sensors, each one measuring a red, green or blue pixel color. In a single-
10   sensor digital camera, however, only a single red, blue or green color value can be determined at a given pixel. In order to obtain the other two missing colors, a technique must be used to estimate or interpolate the missing colors from surrounding pixels in the image. This class of estimation and interpolation techniques is called "demosaicing".
15

The "demosaicing" term is derived from the fact that a color filter array (CFA) is used in front of the image sensors, with the CFA being arranged in a mosaic pattern. This mosaic pattern has only one color value for each of the pixels in the image. In order to obtain the full-color image, the mosaic pattern
20   must be "demosaiced". Thus, demosaicing is the technique of interpolating back the image captured with a mosaic-pattern CFA, so that a full RGB value can be associated with every pixel.

More specifically, a single-sensor digital camera captures the image using
25   an image sensor array that is preceded in the optical path by a CFA. A highly common mosaic CFA is called the Bayer mosaic pattern. The Bayer mosaic pattern (or Bayer filter) is shown in FIG. 1. For each 2x2 set of pixels, two diagonally opposed pixels have green filters, and the other two pixels have red and blue filters. Since the color green (G) carries most of the luminance
30   information for humans, its sampling rate is twice that of the color red (R) and the

color blue (B). The Bayer filter is described in U.S. Patent No. 3,971,065 by Bayer entitled "Color Imaging Array".

There are multitudes of demosaicing techniques available. One of the
5   simplest approaches to demosaicing is bilinear interpolation. In general, bilinear interpolation uses three color planes that are independently interpolated using symmetric bilinear interpolation. This interpolation uses a pixel's nearest neighbors having the same color as the color that is being interpolated. In particular, referring again to FIG. 1, in bilinear interpolation the green value $g(i,j)$
10   at a pixel position (i,j) that falls in a red or blue pixel is computed by the average of the neighboring green values in a cross pattern, as follows,

$$\hat{g}(i,j) = \frac{1}{4} \sum g(i+m, j+n) \tag{1}$$
$$(m,n)=\{(0,-1),\ (0,1),\ (-1,0),\ (1,0)\}$$

Equation (1) corresponds to estimating the green value at the pixel marked '+'
15   (also called the "current pixel") in FIG. 1 as the average of the observed green values marked '·'. It should be noted that the current pixel has a red color, and therefore the green and blue color values need to be interpolated. At image boundaries, only pixels that fall within the image are included, and the scaling adjusted.
20

A graphic illustration of how missing colors are interpolated at a given pixel using bilinear interpolation is shown in FIGS. 2A-H. As shown in FIGS. 2A-H, it should be noted that bilinear techniques typically uses small region of support. The region of support is the size of a pixel neighborhood whose values
25   are considered for the interpolation of any given pixel. As shown in FIGS. 2A-H, the region of support for the bilinear interpolation techniques described below typically is a 3x3 pixel region of support. Using this small of a region of support keeps memory usage and computational complexity to a minimum.

Referring to FIGS. 2A and 2B, illustrated is the bilinear interpolation technique of equation (1) for finding a green color value at a given pixel. In FIG. 2A, a green color value at a red pixel is found by using the green pixels neighboring the red pixel. These neighboring green pixels are marked with a "1"

5   in FIG. 2A. Similarly, in FIG. 2B, a green color value at a blue pixel is found by using the green pixels neighboring the blue pixel. Once again, the green pixels are marked with a "1".

To find missing red (R) or blue (B) colors, equation (1) still applies.

10   However, now a diagonal cross pattern is used. An exception is that for pixel positions (i,j) that fall in a green pixel only two red neighboring pixels are averaged to produce an interpolated red color value, because there are only two immediately neighboring red pixels. The same is also true for finding blue color values. It should be noted that, besides the computational simplicity of equation

15   (1), its output value is guaranteed to have the same dynamic range of the input value, so no overflow detection and control logic is needed on the output value.

More specifically, FIGS. 2C-E illustrate the bilinear interpolation technique of equation (1) for finding a red color value at a given pixel. Specifically, in FIG.

20   2C, a red color value is found at a green pixel in a red row and a blue column by using the two red neighboring pixels. Similarly, In FIG. 2D, a red color value at a green pixel in a blue row and a red column is found by using the two neighboring red pixels in the red column. In FIG. 2E, a red color value at a blue pixel is found using the four neighboring red pixels. In FIGS. 2C-E, each of the neighboring red

25   pixels used in the bilinear interpolation technique are marked with a "1".

Similarly, FIGS. 2F-H illustrate the bilinear interpolation technique of equation (1) for finding a blue color value at a given pixel. Specifically, in FIG. 2F, a blue color value is found at a green pixel in a blue row and a red column by

30   using the two blue neighboring pixels. In FIG. 2G, a blue color value at a green pixel in a red row and a blue column is found by using the two neighboring red

pixels in the blue column. In FIG. 2H, a blue color value at a red pixel is found using the four neighboring blue pixels. In FIGS. 2F-H, each of the neighboring blue pixels used in the bilinear interpolation technique are marked with a "1".

5   One problem, however, with the above-described bilinear interpolation techniques is that they generate significant artifacts in the color image. This is especially true across edges and other high-frequency content in the image, since bilinear interpolation does not consider the statistical correlation among RGB values. For example, referring to FIGS. 2D and 2F, suppose there is a

10   sharp change in pixel intensity values, so that all pixels in the rightmost column are much brighter than the others. Then, the interpolated blue value of the center pixel (from FIG. 2F) will be somewhat bright, because its blue neighbor to the right is very bright. However, the interpolated red value for that same pixel (from FIG. 2D) will not be bright, because its two red neighbors are not very

15   bright. Therefore, in this case there will be a bright interpolated blue pixel and a not so bright interpolated red pixel; the excess blue and lack of red will make the interpolated pixel color be tinted towards cyan, causing a color fringe artifact. Thus, while bilinear interpolation techniques are fast, computationally non-intensive, and easy to implement, they are also notorious for producing low-

20   quality images due to the significant artifacts (mainly blurriness and color fringing) that they generate.

        Better, albeit more complex, interpolation techniques take correlation among RGB values into account. One group of interpolation techniques consider

25   such correlation by using improved linear filters. For example, such a technique is described in a paper by S.-C. Pei and I.-K. Tam entitled "Effective color interpolation in CCD color filter array using signal correlation," in *Proc. ICIP*, pp. 488–491, Sept. 2000. Another group of interpolation techniques consider such correlation by using nonlinear filters. These nonlinear filters essentially adapt

30   interpolation smoothness to a measure of image activity or edginess. For example, these nonlinear interpolation techniques are discussed in the following

papers: (1) P. Longère, X. Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proc. IEEE,* vol. 90, pp. 123–132, Jan. 2002; and (2) R. Ramanath, W. E. Snyder, and G. L. Bilbro, "Demosaicking methods for Bayer color arrays," *J. Electronic Imaging,* vol.

5   11, pp. 306–315, July 2002.

Exploiting correlation among RGB channels is the main idea behind improving demosaicing performance in nonlinear interpolation techniques. Specifically, it can be assumed that in a luminance/chrominance decomposition,

10   the chrominance components do not vary much across pixels. In a constant-hue approach described in U.S. Patent No. 4,724,395 to Freeman entitled, "Median filter for reconstructing missing color samples", the green channel is bilinearly interpolated and then the red and blue channels are interpolated so as to maintain a constant hue, defined as the R/G and B/G ratios. However, one

15   problem with this technique by Freeman is that even at the expense of computing these ratios, the technique still produces visible artifacts. Moreover, using complex operations (such as division and multiplication) in the computing of interpolations greatly increases the computational complexity, processing overhead, and implementation cost.

20

Improved results for nonlinear interpolation techniques can be obtained by starting with bilinearly interpolated green pixels and then applying median filters to the interpolated values of the color differences R – G and B – G. Improved performance can be obtained by using gradient-based nonlinear techniques,

25   which typically estimate edge directions and adjust the interpolation formulas so that filtering is performed preferentially along edge directions, and not across them. For example, one gradient-based nonlinear technique is described in U.S. Patent No. 5,373,322 to C.A. Laroche and M.A. Prescott entitled "Apparatus and method for adaptively interpolating a full color image utilizing chrominance

30   gradients". The Laroche and Prescott technique interpolates the green channel by using both the red and blue channels to determine edge directions, which

determine unequal weights to the terms in equation (1) for the green channel. The color differences R–G and B–G then are interpolated. A technique described in U.S. Patent No. 5,506,619 to J.E. Adams and J.F. Hamilton, Jr., entitled "Adaptive color plane interpolation in a single color electronic cameras"

5 improves on the Laroche and Prescott technique by considering both first and second order pixel differences (see also J. E. Adams, "Design of practical color filter array interpolation algorithms for digital cameras," *Proc. SPIE*, vol. 3028, pp. 117–125, Feb. 1997). These gradient techniques, however, merely compute a gradient and use it to select which interpolation technique to use.

10

A technique described in a paper by E. Chang, S. Cheung, and D. Y. Pan, entitled "Color filter array recovery using a threshold-based variable number of gradients," in *Proc. SPIE*, vol. 3650, pp. 36–43, Jan.1999, is an improvement on the above techniques by considering a variable number of gradients. A simpler

15 but efficient algorithm that uses soft decision rules to combine interpolation results from horizontal and vertical directions is presented in a paper by X. Wu and N. Zhang, entitled "Primary-consistent soft-decision color demosaic for digital cameras," in *Proc. ICIP*, vol. I, pp. 477–480, Sept. 2003.

20 Iterative methods can lead to further improvement by using results from blue and red interpolations to correct the green interpolation, and vice-versa. That is the basis of a technique described in a paper by R. Kimmel, "Demosaicing: image reconstruction from color CCD samples," *IEEE Trans. on Image Processing,* vol. 8, pp. 1221–1228, Sept. 1999. In Kimmel's approach, the

25 interpolation steps are based on a combination of the constant-hue and gradient-based methods. A technique based on iterative projections is presented in a paper by B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, entitled "Color plane interpolation using alternating projections", in *IEEE Trans. on Image Processing,* vol. 11, pp. 997–1013, Sept. 2002. The Gunturk technique has the

30 best performance to date on a popular set of standard test images. One

problem, however with the Gunturk technique is that it has a very high complexity (as many as 480 operations per input pixel).

The above-described nonlinear interpolation techniques typically use a larger region of support that is used in bilinear interpolation techniques. For example, a 5x5 pixel region of support is typical for these nonlinear techniques. A 5x5 region of support yields good correlation between RGB values and produce high-quality images. A smaller region of support greatly reduces the image quality, while larger regions of support require more memory and greatly increase computational complexity.

Therefore, what is needed is an interpolation technique for demosaicing color images that retains the simplicity and speed of the bilinear interpolation technique while producing the high-quality images of more complex techniques, such as nonlinear interpolation techniques. What is also needed is a high-quality interpolation technique that considers correlation among RGB values to improve performance. What is further needed is an interpolation technique that considers this correlation by computing a gradient and uses this computed gradient directly as a correction term and not merely to select an interpolation technique. What is also needed is an interpolation technique that utilizes a larger region of support than bilinear techniques (where the region of support is comparable to that used in nonlinear techniques) for improved quality. What is further needed is an interpolation technique that requires only simple mathematical operations (such as addition) when computing interpolations to increase speed, avoid computational complexity, and allow inexpensive implementation.

SUMMARY OF THE INVENTION

The invention disclosed herein includes a high-quality linear interpolation method and system for demosaicing of digital color images. The present invention overcomes the above-mentioned deficiencies of current interpolation techniques by providing a high-quality interpolation that is simple to implement

and cost efficient. Namely, the gradient-corrected linear interpolation method and system disclosed herein uses color information from a current pixel to obtain a correction term. This correction term then is linearly combined with a prior art interpolation to provide an improved estimate of a missing color at the current

5   pixel. The gradient-corrected linear interpolation method and system have a better performance and lower complexity than current prior art linear demosaicing techniques. In addition, the gradient-corrected linear interpolation method and system outperform many prior art nonlinear demosaicing techniques having a much greater complexity.

10

The gradient-corrected linear interpolation method and system can operate on color images captured by digital cameras. Prior to display, the image is demosaiced using the gradient-corrected linear interpolation method and system. Alternatively, a captured digital color image can be stored in a

15   computing device in a compressed mosaiced form. If an application on the computing device needs to access the image, the application typically calls an appropriate application programming interface (API) to obtain the decompressed data. Most applications expect data in the RGB format, so the demosaicing must be performed before the image data is sent to the requesting application.

20

The gradient-corrected linear interpolation method includes computing an interpolation of a desired color at a current pixel having a current color, where the desired color is used in the interpolation. Next, a correction term is computed using the current color. The correction term then is linearly combined with the

25   interpolation to obtain a corrected interpolation of the desired color at the current pixel. Preferably, in order to simplify computation and preserve computing resources, the interpolation is a prior art bilinear interpolation (such as the one described in the background). However, the interpolation may be virtually any one of the available prior art demosaicing techniques, including linear and

30   nonlinear interpolations.

The correction term is preferably a gradient correction term computed from the current color of the current pixel. For example, assume the current pixel has a image sensor that is capable of determining a green color value for the pixel, and a red color value at the green pixel is desired, as in FIGS. 2C and 2D.

5 The gradient-corrected linear interpolation method computes a gradient at the green pixel by using the green value at that pixel and the values of neighboring green pixels. A fixed portion (preferably controlled by a gain factor) of that gradient value then is added to the interpolated red value (preferably, a bilinearly interpolated red value). Although some prior art demosaicing techniques do

10 compute gradients, they use the gradient to control how the interpolation is computed. On the other hand, the gradient-corrected linear interpolation method disclosed herein uses the gradient directly as a correction term. In other words, some prior art demosaicing techniques compute a gradient and use the gradient to choose an interpolation method based on the gradient, while the gradient-

15 corrected linear interpolation method uses the gradient directly to affect and correct the value produced by a prior art demosaicing technique.

The gradient correction term is linearly combined with the interpolation. Preferably, the gradient correction term is added to the interpolation to improve

20 the estimate of the desired color. The prior art demosaicing techniques that do use gradients typically compute and use complex operators based on the gradients, such as nonlinear filters, "if' statements, and division operations. The gradient-corrected linear interpolation method simplifies computation by using a linear operation to linearly combining the gradient correction term with the

25 interpolation, where the linear operation preferably is addition.

The gradient-corrected linear interpolation method also includes applying a gradient-correction gain to the gradient correction term. This gradient-correction gain affects the amount of gradient correction that is applied to the

30 interpolation. For example, if the gradient-correction gain is 0.8, then only 80% of the gradient correction is linearly combined with the interpolation. The

gradient-correction gain can be selected in a variety of ways. The optimal gradient-correction gain is computed by minimization a mean-squared error. An approximation to this optimal value allows computations using integer arithmetic, no division operations, or both. The gradient-correction may also be selected

5   based on characteristics of the color image sensor or characteristics of an entire digital camera system. The gradient-correction gain may also be varied within the image, if so desired, based on statistics of the color image. These statistics may include the global statistics of the image, the local statistics of the image, or both.

10

The gradient-corrected linear interpolation method includes defining a region of support as the size of a pixel neighborhood that immediately surrounds the current pixel. The region of support then is selected to include pixels nearest the current pixel having the current color and using the region of support to

15   compute the interpolation and the correction term. Preferably, the region of support includes a 5x5 pixel region that is centered at the current pixel. Alternatively, the region of support may be greater than a 5x5 pixel region. There also may be more than one region of support. For example, a first region of support may be used to compute the interpolation, while a second region of

20   support may be used to compute the correction term.

A gradient-corrected linear interpolation system uses the method described above to produce a high-quality interpolation of missing color at a given pixel in a quick, simple and efficient manner. The gradient-corrected linear

25   interpolation system includes a current pixel selector, that selects a current pixel having a current color, and an interpolation module, which computes a first interpolation of color value that is missing and desired at the current pixel. The gradient-corrected linear interpolation system also includes a correction term module, that computes a correction term for the interpolation, and a linear

30   combination module that linearly combines the interpolation and correction term to obtain the desired (or missing) color value at the current pixel.

The correction term computation module further includes a gradient-correction selector that selects the amount of correction that will be applied. Moreover, the correction term computation module also includes a region of

5   support module, which defines a region of support around the current pixel, and a gradient correction module that computes a gradient correction using the current pixel. In addition, correction term computation module includes a gradient-correction gain application module that applies the gain to the computed gradient correction such that an adjusted gradient correction is output.

10

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention can be further understood by reference to the following description and attached drawings that illustrate aspects of the invention. Other features and advantages will be apparent from the following

15   detailed description of the invention, taken in conjunction with the accompanying drawings, which illustrate, by way of example, the principles of the present invention.

Referring now to the drawings in which like reference numbers represent

20   corresponding parts throughout:

FIG. 1 illustrates a typical Bayer mosaic pattern for color image capture in single image sensor digital camera systems.

FIG. 2A illustrates a bilinear interpolation technique for finding a green

25   color value at a red pixel.

FIG. 2B illustrates a bilinear interpolation technique for finding a green color value at a blue pixel.

FIG. 2C illustrates a bilinear interpolation technique for finding a red color value at a green pixel in a red row and a blue column.

30   FIG. 2D illustrates a bilinear interpolation technique for finding a red color value at a green pixel in a blue row and a red column.

FIG. 2E illustrates a bilinear interpolation technique for finding a red color value at a blue pixel.

FIG. 2F illustrates a bilinear interpolation technique for finding a blue color value at a green pixel in a blue row and a red column.

5   FIG. 2G illustrates a bilinear interpolation technique for finding a blue color value at a green pixel in a red row and a blue column.

FIG. 2H illustrates a bilinear interpolation technique for finding a blue color value at a red pixel.

FIG. 3A is a block diagram illustrating a first exemplary implementation of

10   the gradient-corrected linear interpolation system and method disclosed herein.

FIG. 3B is a block diagram illustrating a second exemplary implementation of the gradient-corrected linear interpolation system and method disclosed herein.

FIG. 4 is a general flow diagram illustrating the general operation of the

15   gradient-corrected linear interpolation system shown in FIG. 3.

FIG. 5 is a general flow diagram illustrating the gradient correction computation process.

FIG. 6A illustrates the details of the gradient-corrected linear interpolation method for finding a green color value at a red pixel.

20   FIG. 6B is a detailed diagram illustrating a 5x5 region of support for finding a green color value at a red pixel, as shown in FIG. 6A.

FIG. 6C illustrates the details of the gradient-corrected linear interpolation method for finding a green color value at a blue pixel.

FIG. 6D is a detailed diagram illustrating a 5x5 region of support for finding

25   a green color value at a blue pixel, as shown in FIG. 6C.

FIG. 7A illustrates the details of the gradient-corrected linear interpolation method for finding a red color value at a green pixel located in a red row and a blue column.

FIG. 7B is a detailed diagram illustrating a 5x5 region of support for finding

30   a red color value at a green pixel, as shown in FIG. 7A.

FIG. 7C illustrates the details of the gradient-corrected linear interpolation method for finding a red color value at a green pixel located in a blue row and a red column.

FIG. 7D illustrates the details of the gradient-corrected linear interpolation

5   method for finding a red color value at a blue pixel.

FIG. 7E is a detailed diagram illustrating a 5x5 region of support for finding a red color value at a blue pixel, as shown in FIG. 7D.

FIG. 8A illustrates the details of the gradient-corrected linear interpolation method for finding a blue color value at a green pixel located in a blue row and a

10   red column.

FIG. 8B is a detailed diagram illustrating a 5x5 region of support for finding a blue color value at a green pixel, as shown in FIG. 8A.

FIG. 8C illustrates the details of the gradient-corrected linear interpolation method for finding a blue color value at a green pixel located in a red row and a

15   blue column.

FIG. 8D illustrates the details of the gradient-corrected linear interpolation method for finding a blue color value at a red pixel.

FIG. 8E is a detailed diagram illustrating a 5x5 region of support for finding a blue color value at a red pixel, as shown in FIG. 8D.

20   FIG. 9 illustrates an example of a suitable computing system environment in which the gradient-corrected linear interpolation method and system shown in FIG. 3 may be implemented.

FIG. 10 is a block diagram illustrating the details of the gradient-corrected linear interpolation system shown in FIG. 3.

25   FIG. 11 is a block diagram illustrating the details of the correction term computation module shown in FIG. 10.

DETAILED DESCRIPTION OF THE INVENTION

In the following description of the invention, reference is made to the

30   accompanying drawings, which form a part thereof, and in which is shown by way of illustration a specific example whereby the invention may be practiced. It

is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

I.   Introduction

5   Nearly all current digital cameras use a single sensor for each pixel in a color image. As a result, for each pixel, only one of the three red, green and blue (RGB) colors is captured. In order to produce the color image, however, each RGB color value is needed at every pixel. Therefore, the cameras use a technique called "demosaicing" to generate the missing color values at every
10   pixel.

       Unfortunately, the techniques for interpolating the missing RGB data are a tradeoff between complexity and quality. For example, bilinear interpolation techniques use simple computational operations, linear operators, and small
15   regions of support (such as a 3x3 region of support). This makes these bilinear interpolation techniques fast, computationally non-intensive, and easy to implement. However, these bilinear techniques also generate significant artifacts in the color image, which can severely degrade image quality. On the other hand, some nonlinear interpolation techniques produce noticeably improved
20   image quality, but are significantly slower, use more processing power, use complex computational operations, and are harder to implement.

       The gradient-corrected linear interpolation method and system disclosed herein includes a new interpolation technique for demosaicing of color images.
25   The method and system take an interpolation computed by a current technique, compute a correction term (such as a gradient of the desired color at a given pixel), and linearly combine the interpolation and the correction term. The method and system keep the computational complexity to a minimum by performing fixed linear operations with low-cost operators (such as additions and
30   shifts). Moreover, by directly using a gradient as a correction term and not just a means of choosing an interpolation method, the gradient-corrected linear

interpolation method and system simplifies implementation by avoiding the need for complex programming (such as numerous "if" statements, which slow execution). Image quality is improved because the gradient considers correlation among RGB values. The gradient-corrected linear interpolation method and

5   system also utilizes a larger region of support, similar to the nonlinear techniques, to improve estimates of missing color values.

The gradient-corrected linear interpolation method and system can lead to an improvement in peak-signal-to-noise ratio (PSNR) of over 5.5 dB when

10   compared to bilinear interpolation techniques, and about 0.7 dB improvement in red (R) and blue (B) interpolation when compared to some current linear interpolation techniques. The gradient-corrected linear interpolation method and system also outperforms most nonlinear interpolation techniques, without the artifacts due to nonlinear processing, and at a greatly reduced computational

15   complexity.

II.   General Overview

FIGS. 3A and 3B are block diagrams illustrating two exemplary implementations of the gradient-corrected linear interpolation system and method

20   disclosed herein. It should be noted that FIGS. 3A and 3B are merely two of several ways in which the gradient-corrected linear interpolation system and method may implemented and used.

Referring to FIG. 3A, in the first exemplary implementation, the gradient-

25   corrected linear interpolation system 300 resides on a digital camera system 310. In general, the gradient-corrected linear interpolation system 300 uses the gradient-corrected linear interpolation method to process image data that is obtained from an image sensor array 315 and outputs missing color values for each pixel 320. More specifically, as shown in FIG. 3A, the gradient-corrected

30   linear interpolation system 300 is part of the digital camera system 310 and includes the image sensor array 315. The array 315 is made up of a plurality of

image sensors (shown by the circles in FIG. 3A), where each image sensor is capable of determining a single color value.

5   The image sensor array 315 includes a sensor substrate 330, a color filter array (CFA) 340, and a microlens array 350. It should be noted that typically the microlens array 350 is disposed over the CFA 340, which in turn is disposed over the substrate 330, as indicated in FIGS. 3A and 3B by the dashed lines. The light from an image to be captured (not shown) enters the digital camera system 310 (as shown by the arrows 355). The purpose of the microlens array 350 is to

10   focus the incoming light 355 onto each pixel (represented by a single image sensor). The CFA 340 changes the color response of each pixel. The sensor substrate 330 captures the specific color incoming light (as dictated by the CFA 340) and generates an electrical signal. In this manner, the image sensor array 315 samples the incoming light 355 from an image with red, green and blue

15   sensors arranged in the image sensor array 315.

       Each pixel in the image sensor array 315 is the output of a red, green or blue sensor. Thus, the sensor output is an electrical signal from each sensor that corresponds to a single color value for each pixel in the image sensor array 315.

20   The output is raw, mosaic-patterned pixel data 360. As mentioned above, in order to display the full color image, color information for red, green and blue colors must be obtained at each pixel. This is achieved by demosaicing, or the interpolating the missing color values in the mosaic-patterned pixel data 360.

25   The gradient-corrected linear interpolation system 300 use a novel demosaicing technique to interpolate a high-quality estimate of the missing color values. The gradient-corrected linear interpolation system 300 resides on a processing device 370 located on the digital camera system 310. As detailed below, the gradient-corrected linear interpolation system 300 uses the gradient-

30   corrected linear interpolation method to obtain the missing color values at each pixel 320 in the image sensor array 315.

Referring now to FIG. 3B, in the second exemplary implementation the gradient-corrected linear interpolation system 300 resides on a computing device 380. The raw mosaic-patterned pixel data 360 is sent from the digital camera system 310 to the computing device 380. This pixel data 360 typically is sent to the computing device 380 via an interface connection 385, which can be a USB interface, an IEEE 1394 interface, a Bluetooth® interface, or any other interface suitable for carrying the raw pixel data 360 from the digital camera system 310 to the computing device 380. As detailed below, the gradient-corrected linear interpolation system 300 uses the gradient-corrected linear interpolation method to obtain the missing color values at each pixel 390 for use by the computing device 380.

III.   Operational Overview

The operation of the gradient-corrected linear interpolation system 300 and method used therein shown in FIG. 3 now will be discussed. FIG. 4 is a general flow diagram illustrating the general operation of the gradient-corrected linear interpolation system shown in FIG. 3. The gradient-corrected linear interpolation method begins by inputting a current pixel from a color image (box 400). The current pixel has an associated current color. Namely, the current color is the color that the image sensor of the current pixel is filtered to receive. Next, an interpolation is computed for a desired color at the current pixel (box 410). In other words, any of the missing colors at the current pixel are interpolated or estimated. By way of example, if the current color is green, then the desired color may be red or blue. The interpolation is a prior art interpolation technique, using any of the above-mentioned interpolation techniques including linear, bilinear, nonlinear, bi-cubic and Lanczos interpolation techniques. In a preferred embodiment, a bilinear interpolation technique is used. Moreover, the desired color is used to compute the interpolation. Neighboring pixels of the desired color generally are used in computing the interpolation.

A correction term then is computed (box 420). This computation is performed using the current color of the current pixel. The current pixel typically is used in computing the correction term, along with neighboring pixels of the current pixel having the current color. In a preferred embodiment, the correction

5   term is a gradient correction, obtained by computing a gradient of the current pixel, as discussed in detail below.

The gradient-corrected linear interpolation method and system include the novel criterion that edges have a much stronger luminance than chrominance

10   components. Thus, when computing an interpolation of a green color value at a red pixel, the red color value at that location is not thrown away. Instead, the present invention utilizes this valuable information to compute the correction term. In contrast, prior art interpolation techniques throw this information away. In this example, the red color value would be compared to its interpolated value

15   using a prior art interpolation technique for the neighboring red color values. If it is different from that estimate, it likely means that there is a sharp luminance change at that pixel. The present invention then will correct the interpolated green color value by adding at least a portion of the correction term (corresponding to the estimated luminance change). Such comparisons are

20   implicitly performed by the gradient-based correction terms, so that no "if" statements or conditional branch logic are necessary.

Finally, the interpolation and the correction term are linearly combined to obtain a corrected interpolation of the desired color at the current pixel (box 430).

25   In a preferred embodiment, interpolation and the correction term are added together to obtain the corrected interpolation. Moreover, a gradient-correction gain may be applied to the gradient correction to dictate the amount (or portion or percentage) of the gradient correction that is linearly combined with the interpolation. As explained below, this gradient-correction gain may be selected

30   using a variety of techniques.

Unlike prior art interpolation techniques, when the gradient-corrected linear interpolation method computes the correction term for an interpolation of a green value at a red pixel location, for example, the red value at that location is not discarded. The current color (red value) is valuable information! Instead, the

5   gradient-corrected linear interpolation method compares that red value to its interpolation (or estimate) for the nearest red samples. If the correction term is different from that estimate, it likely indicates there is a sharp luminance change at the current pixel. Thus, the gradient-corrected linear interpolation method corrects a prior art interpolation at the current pixel (having a green color in this

10   example) by linearly combining at least a portion of the correction term to the interpolation, where the correction term represents the this estimated luminance.

IV.   Operational Details and Working Example

FIG. 5 is a general flow diagram illustrating the gradient correction

15   computation process. In particular, a gradient-correction gain is determined (box 500). As detailed below, the gain may be determined in a variety of ways. Next, a region of support is defined around a current pixel in a color image (box 510). This region of support is defined as a size of a pixel neighborhood whose color values are considered for computations associated with any given pixel, such as

20   the current pixel. These computations includes computing an interpolation and computing a correction term (such as a gradient).

A gradient correction then is computed (box 520). This gradient is computed using the current pixel (having a current color) and pixels within the

25   region of support also having the current color. The gradient-correction gain then is applied to the gradient correction (box 530). It should be noted that each of the gradient-correction gains is associated with a particular interpolation pattern. Applying the gain to the correction dictates how much of the gradient correction is used to correct the interpolation. Finally, an adjusted gradient correction is

30   output (box 540). This adjusted gradient correction may be linearly combined with the interpolation in order to improve the quality of the interpolation alone.

In order to more fully understand the gradient-corrected linear interpolation method and system disclosed herein, the operational details of an exemplary working example are presented. It should be noted that this working example is

5 only one way in which the gradient-corrected linear interpolation method and system may be implemented.

Gradient Correction

The following formulas are presented to provide a corrected interpolation

10 at a pixel location using a gradient correction. In order to illustrate the details of the gradient-corrected linear interpolation method and system, the interpolation presented in a bilinear interpolation technique. However, it should be noted that any other type of interpolation technique may be used.

15 Specifically, to interpolate green color values (G) at a red pixel (R) location, the following formula is used:

$$\hat{g}(i,j) = \hat{g}_B(i,j) + \alpha \Delta_R(i,j) \qquad (2)$$

where,

20 $$\hat{g}(i,j)$$

is the gradient-corrected linear interpolation,

$$\hat{g}_B(i,j)$$

is the bilinear interpolation,

25

$$\alpha$$

is the gradient-correction gain for that particular interpolation pattern, and

$$\Delta_R(i,j)$$

30 is the gradient of red (R) at that pixel location, computed by

$$\Delta_R(i, j) \triangleq r(i, j) \; - \; \frac{1}{4} \sum r(i+m, j+n) \tag{3}$$

$$(m,n)=\{(0,-2), (0,2), (-2,0), (2,0)\}$$

Thus, the gradient-corrected linear interpolation method and system
5   corrects the bilinear interpolation estimate by a measure of the gradient Δ_R for the known color at the pixel location, which is estimated by the simple formula in equation (3). The gradient-correction gain α controls the intensity of such correction. The gradient-corrected linear interpolation method and system is in fact a gradient-corrected bilinear interpolated approach, with a gain parameter to
10   control how much correction is applied.

For interpolating green (G) at blue (B) pixels, the same formula is used, but corrected by $)_B(i,j)$. For interpolating red (R) at green (G) pixels, the following formula is used:

15

$$\hat{r}(i, j) = \hat{r}_B(i, j) + \beta \Delta_G(i, j) \tag{4}$$

with $\Delta_G(i,j)$ determined by a 9-point region of support, as detailed below.
20

For interpolating red (R) at blue (B) pixels, the following formula is used:

$$\hat{r}(i, j) = \hat{r}_B(i, j) + \gamma \Delta_B(i, j) \tag{5}$$

25

with $\gamma_B(i,j)$ computed on a 5-point region of support, as also detailed below.

The formulas for interpolating blue (B) are similar, by symmetry.

30   <u>Gradient-Correction Gains</u>

In order to determine appropriate values for the gradient-correction gains (α, β, and γ, one for each of the interpolation patterns, as discussed above), a Wiener approach was used. In other words, the values that led to minimum mean-square error interpolation were computed. This computation was

5   performed using second-order statistics computed from a typical image data set. Next, the optimal Wiener coefficients were approximated by integer multiples of small powers of ½. The final results of the optimal gain values were: $\alpha = 1/2$, $\beta = 5/8$, and $\gamma = ¾$. From the values of $\{\alpha,\beta,\gamma\}$, the equivalent linear finite impulse response (FIR) filter coefficients for each interpolation case can be computed.

10   The resulting coefficient values make the filters quite close (within 5% in terms of mean-square error) to the optimal Wiener filters for a 5x5 region of support.

Thus, it is believed that the only way to design a practical linear filter with a meaningfully lower mean-square interpolation error would be to use larger

15   regions of support. This would not only increase computational complexity and memory required, but would also lead to ringing artifacts around the edges within the image. One way to evaluate computational complexity is to count the number of nonzero filter coefficients within the 5x5 regions of support. For example, in the paper by S.-C. Pei and I.-K. Tam (cited above), there are 9

20   nonzero coefficients for the green (G) channel, and an average of 13 each for the red (R) and blue (B) channels. In contrast, the gradient-corrected linear interpolation method and system has a slightly lower complexity, namely, 9 coefficients for the G channel and 11 coefficients each for the R and B channels.

25   FIGS. 6A-D are detailed diagrams illustrating the gradient-corrected linear interpolation method for finding green color values. In FIG. 6A, a green color value is to be found at a red pixel 600. It should be noted that the parameter "a" is related to the gradient-correction gain α by $a = 2\alpha$, as discussed below. As shown in FIG. 6A, an interpolation (such as a bilinear interpolation) is performed

30   using the green pixels surrounding the red pixel 600. In addition, the gradient

correction is computed using the red pixels surrounding the red pixel 600 as well as the red pixel 600 itself.

FIG. 6B is a detailed diagram illustrating a 5x5 region of support for finding a green value at a red pixel, as shown in FIG. 6A. In particular, the gradient-corrected linear interpolation method for the 5x5 region of support shown in FIG. 6B includes:

1.   Start with bilinearly-interpolated Green:

$$\hat{G}_{33B} = \frac{G_{23} + G_{32} + G_{34} + G_{43}}{4}$$

2.   Compute Red gradient:

$$\Delta_{33}^{R} = R_{33} - \frac{R_{13} + R_{31} + R_{35} + R_{53}}{4}$$

3.   Apply correction:

$$\hat{G}_{33new} = \hat{G}_{33B} + \alpha \Delta_{33}^{R}$$

where α is a "correction gain", with $0 < \alpha < 1$. A typical value of $\alpha = \frac{1}{2}$.

Note that the formulas above can be combined in a general FIR-filter-like formula

$$\hat{G}_{33new} = \frac{G_{23} + G_{32} + G_{34} + G_{43}}{4} + \alpha \left(R_{33} - \frac{R_{13} + R_{31} + R_{35} + R_{53}}{4} \right)$$

$$= \frac{1}{8} \left[2 \left(G_{23} + G_{32} + G_{34} + G_{43} \right) + 4aR_{33} - a \left(R_{13} + R_{31} + R_{35} + R_{53} \right) \right]$$

where the parameter a is directly related to the gain α by a = 2α, so for α = ½ we have a = 1. In this form, it is clear that the final estimated green (G) value can be

computed as a linear combination of the denoted green and red values, using only additions and shifts (since multiplication by 2 and 4 can be performed by one-position and two-position left shifts, respectively, and division by eight can be performed by a three-position right shift).

5

In FIG. 6C, a green color value is to be found at a blue pixel 610. As shown in FIG. 6C, an interpolation (such as a bilinear interpolation) is performed using the green pixels surrounding the blue pixel 610. In addition, the gradient correction is computed using the blue pixels surrounding the blue pixel 610 as

10   well as the blue pixel 610 itself.

FIG. 6D is a detailed diagram illustrating a 5x5 region of support for finding a green value at a blue pixel, as shown in FIG. 6C. In particular, the gradient-corrected linear interpolation method for the 5x5 region of support shown in FIG.

15   6C includes:

1.   Start with bilinearly-interpolated Green:

$$\hat{G}_{33B} = \frac{G_{23} + G_{32} + G_{34} + G_{43}}{4}$$

20

2.   Compute Blue gradient:

$$\Delta_{33}^{B} = B_{33} - \frac{B_{13} + B_{31} + B_{35} + B_{53}}{4}$$

25   3.   Apply correction:

$$\hat{G}_{33new} = \hat{G}_{33B} + \alpha \Delta_{33}^{B}$$

where α is a "correction gain", with $0 < \alpha < 1$. A typical value

30   of $\alpha = \frac{1}{2}$.

As before, the formulas above can be combined in a general FIR-filter-like formula

$$\hat{G}_{33new} = \frac{G_{23}+G_{32}+G_{34}+G_{43}}{4} + \alpha\left(B_{33} - \frac{B_{13}+B_{31}+B_{35}+B_{53}}{4}\right)$$

$$= \frac{1}{8}\left[2\left(G_{23}+G_{32}+G_{34}+G_{43}\right) + 4aB_{33} - a\left(B_{13}+B_{31}+B_{35}+B_{53}\right)\right]$$

where again the parameter a is directly related to the gain α by a = 2α, so for α =

5   ½ we have a = 1. Again, it is clear that the above filtering operation can be efficiently computed using only additions and shifts.

FIGS. 7A-E are detailed diagrams illustrating the gradient-corrected linear interpolation method for finding red color values. In FIG. 7A, a red color value is

10   found at a green pixel 700. The green pixel 700 is located in a red row and a blue column. It should be noted that the parameter "b" is related to the gradient-correction gain β by b = (8/5)β, as discussed below. As shown in FIG. 7A, an interpolation (such as a bilinear interpolation) is performed using the two red pixels horizontally neighboring the green pixel 700. Moreover, the gradient

15   correction is computed using the green pixels surrounding the green pixel 700 as well as the green pixel 700 itself.

FIG. 7B is a detailed diagram illustrating a 5x5 region of support for finding a red value at a green pixel, as shown in FIG. 7A. In particular, the gradient-

20   corrected linear interpolation method for the 5x5 region of support shown in FIG. 7A includes:

1.   Start with bilinearly-interpolated Red:

25

$$\hat{R}_{33B} = \frac{R_{32}+R_{34}}{2}$$

2.   Compute Green gradient:

$$\Delta_{33}^G = G_{33} - \frac{G_{22} + G_{24} + G_{31} + G_{35} + G_{42} + G_{44} - (G_{13} + G_{53})/2}{5}$$

3.   Apply correction:

$$\hat{R}_{33new} = \hat{R}_{33B} + \beta \Delta_{33}^G$$

5

where β is a "correction gain", with $0 < \beta < 1$. A typical value
is $\beta = 5/8$.

10   Note that the formulas above can be combined in a general FIR-filter-like
formula

$$\hat{R}_{33new} = \frac{R_{32} + R_{34}}{2} + \beta \left(G_{33} - \frac{G_{22} + G_{24} + G_{31} + G_{35} + G_{42} + G_{44} - (G_{13} + G_{53})/2}{5} \right)$$

$$= \frac{1}{8}\left[4(R_{32} + R_{34}) + 5bG_{33} - b(G_{22} + G_{24} + G_{31} + G_{35} + G_{42} + G_{44}) - (b/2)(G_{13} + G_{53}) \right]$$

where the parameter b is directly related to the gain β by b = (8/5)β, so for β = 5/8
15   we have b = 1. In this form, it is clear that the final estimated red (R) value can
be computed as a linear combination of the denoted red and green values, using
only additions and shifts (e.g. multiplication by 5 can be performed by a two-
position left shift and an addition).

20   In FIG. 7C, a red color value is to be found at a green pixel 710. In this
case, however, the green pixel 710 is located in a blue row and a red column. As
shown in FIG. 7C, an interpolation (such as a bilinear interpolation) is performed
using the two red pixels vertically neighboring the green pixel 710. Similar to the
situation shown in FIGS. 7A and 7B, the gradient correction is computed using
25   the green pixels surrounding the green pixel 710 as well as the green pixel 710
itself.

In FIG. 7D, a red color value is to be found at a blue pixel 720. It should be noted that the parameter "c" is related to the gradient-correction gain γ by $c = 2\gamma$, as discussed below. As shown in FIG. 7D, an interpolation (such as a bilinear interpolation) is performed using the four red pixels neighboring the blue

5   pixel 720. Moreover, the gradient correction is computed using the blue pixels surrounding the blue pixel 720 as well as the blue pixel 720 itself.

FIG. 7E is a detailed diagram illustrating a 5x5 region of support for finding a red value at a blue pixel, as shown in FIG. 7D. In particular, the gradient-

10   corrected linear interpolation method for the 5x5 region of support shown in FIG. 7D includes:

1.   Start with bilinearly-interpolated Red:

15

$$\hat{R}_{33B} = \frac{R_{22} + R_{24} + R_{42} + R_{44}}{4}$$

2.   Compute Blue gradient:

$$\Delta_{33}^{B} = B_{33} - \frac{B_{13} + B_{31} + B_{35} + B_{53}}{4}$$

20

3.   Apply correction:

$$\hat{R}_{33new} = \hat{R}_{33B} + \gamma \Delta_{33}^{B}$$

25   where γ is a "correction gain", with $0 < \gamma < 1$. A typical value is $\gamma = \frac{3}{4}$.

Note that the formulas above can be combined in a general FIR-filter-like formula

$$\hat{R}_{33new} = \frac{R_{22} + R_{24} + R_{42} + R_{44}}{4} + \gamma\left(B_{33} - \frac{B_{13} + B_{31} + B_{35} + B_{53}}{4}\right)$$

$$= \frac{1}{8}\left[2\left(R_{22} + R_{24} + R_{42} + R_{44}\right) + 4cB_{33} - c\left(B_{13} + B_{31} + B_{35} + B_{53}\right)\right]$$

where the parameter c is directly related to the gain γ by c = 2γ, so for γ = 3/4 we have c = 3/2. In this form, it is clear that the final estimated red (R) value can be computed as a linear combination of the denoted red and blue values, using only

5   additions and shifts (e.g. multiplication by 3/2 can be performed by a one-position right shift and an addition).

FIGS. 8A-E are detailed diagrams illustrating the gradient-corrected linear interpolation method for finding blue color values. In FIG. 8A, a blue color value

10   is found at a green pixel 800. The green pixel 800 is located in a blue row and a red column. It should be noted that the parameter "b" is related to the gradient-correction gain β by b = (8/5)β, as discussed above. As shown in FIG. 8A, an interpolation (such as a bilinear interpolation) is performed using the two blue pixels horizontally neighboring the green pixel 800. In addition, the gradient

15   correction is computed using the eight green pixels surrounding the green pixel 800 as well as the green pixel 800 itself.

FIG. 8B is a detailed diagram illustrating a 5x5 region of support for finding a blue value at a green pixel, as shown in FIG. 8A. In particular, the gradient-

20   corrected linear interpolation method for the 5x5 region of support shown in FIG. 8A includes:

1.   Start with bilinearly-interpolated Blue:

$$\hat{B}_{33B} = \frac{B_{32} + B_{34}}{2}$$

25

2.   Compute Green gradient:

$$\Delta_{33}^{G} = G_{33} - \frac{G_{22} + G_{24} + G_{31} + G_{35} + G_{42} + G_{44} - (G_{13} + G_{53})/2}{5}$$

        3.   Apply correction:

$$\hat{B}_{33new} = \hat{B}_{33B} + \beta \Delta_{33}^{G}$$

where β is a "correction gain", with $0 < \beta < 1$. A typical value is $\beta = 5/8$.

Note that the formulas above can be combined in a general FIR-filter-like formula

$$\hat{B}_{33new} = \frac{B_{32} + B_{34}}{2} + \beta \left(G_{33} - \frac{G_{22} + G_{24} + G_{31} + G_{35} + G_{42} + G_{44} - (G_{13} + G_{53})/2}{5} \right)$$

$$= \frac{1}{8} \left[4(B_{32} + B_{34}) + 5bG_{33} - b(G_{22} + G_{24} + G_{31} + G_{35} + G_{42} + G_{44}) - (b/2)(G_{13} + G_{53}) \right]$$

where the parameter b is directly related to the gain β by b = (8/5)β, so for β = 5/8 we have b = 1, as discussed above.

In FIG. 8C, a blue color value is to be found at a green pixel 810. In this case, however, the green pixel 810 is located in a red row and a blue column. As shown in FIG. 8C, an interpolation (such as a bilinear interpolation) is performed using the two blue pixels vertically neighboring the green pixel 810. Similar to the situation shown in FIGS. 8A and 8B, the gradient correction is computed using the eight green pixels surrounding the green pixel 810 as well as the green pixel 810 itself.

In FIG. 8D, a blue color value is to be found at a red pixel 820. It should be noted that the parameter "c" is related to the gradient-correction gain γ by c = 2γ, as discussed above. As shown in FIG. 8D, an interpolation (such as a bilinear interpolation) is performed using the four blue pixels neighboring the red

pixel 820. In addition, the gradient correction is computed using the four red pixels surrounding the red pixel 820 as well as the red pixel 820 itself.

FIG. 8E is a detailed diagram illustrating a 5x5 region of support for finding a blue value at a red pixel, as shown in FIG. 8D. In particular, the gradient-corrected linear interpolation method for the 5x5 region of support shown in FIG. 8D includes:

1.   Start with bilinearly-interpolated Red:

$$\hat{B}_{33B} = \frac{B_{22} + B_{24} + B_{42} + B_{44}}{2}$$

2.   Compute Red gradient:

$$\Delta_{33}^{R} = R_{33} - \frac{R_{13} + R_{31} + R_{35} + R_{53}}{4}$$

3.   Apply correction:

$$\hat{B}_{33new} = \hat{B}_{33B} + \gamma \Delta_{33}^{R}$$

where γ is a "correction gain", with $0 < \gamma < 1$. A typical value is $\gamma = \frac{3}{4}$.

Note that the formulas above can be combined in a general FIR-filter-like formula

$$\hat{B}_{33new} = \frac{B_{22} + B_{24} + B_{42} + B_{44}}{2} + \gamma\left(R_{33} - \frac{R_{13} + R_{31} + R_{35} + R_{53}}{4}\right)$$

$$= \frac{1}{8}\left[2(B_{22} + B_{24} + B_{42} + B_{44}) + 4cR_{33} - c(R_{13} + R_{31} + R_{35} + R_{53})\right]$$

where the parameter c is directly related to the gain γ by c = 2γ, so for γ = 3/4 we have c = 3/2. In this form, it is clear that the final estimated red (R) value can be

computed as a linear combination of the denoted blue and red values, using only additions and shifts, as discussed above.

It should be noted that there are three types of interpolation patterns, so

5   there are three types of gradient-correction gains (α, β, γ). These gains may be selected in a variety of ways. One way is to use the optimal gain values. In general, these optimal values are $\alpha = 1/2$, $\beta = 5/8$, and $\gamma = 3/4$. The differences between the optimal gain values are due to the differences in correlation when looking at the different patterns.

10

In actuality, the optimal gain values are not exactly 1/2, 5/8, and 3/4, respectively. They are close, but for implementation purposes they are approximated by the nearest number that can be a rational number where the denominator is a power of 2. This simplifies operations and eliminates the need

15   for floating point arithmetic, thus greatly simplifying computational complexity. Division by 8 is merely a shift of 3 bits to the right, so that computations can be done in integer arithmetic. Thus, this particular selection of gains has to do with ensuring that computations are performed in integer arithmetic with no division.

20   Another way to pick the gains is based dependent on the digital camera manufacturer. The lens of most digital cameras usually performs some optical low-pass filtering (or blurring). This is because manufacturers know that demosaicing will produce some color fringes, and if they blur the image slightly before it hits the sensor the fringes caused by the demosaicing will be

25   attenuated.

Each manufacturer uses its own color matrix (even though it is a Bayer matrix, they often use different types of color filtering). This means that the frequency response will look slightly different between the manufacturers. For

30   example, with a first manufacturer's filter, there may be less correlation between the blue and the red than a filter from another manufacturer. Given the

frequency response and correlation statistics for each different manufacturer's cameras, the gains can be fine-tuned based on these statistics. Moreover, these gains could be reapproximated to simpler values, such as values that allow integer arithmetic and no division.

5

The gains can also be selected in an adaptive manner. In adaptive gain selection, the gains vary based on statistics of the image. The statistics may be global or local. Global image statistics are based on the image as a whole. Local image statistics are based not only on the entire image, but also on specific

10   regions within the image. From such statistics better inter-channel correlation values can be computed, so the corresponding new optimal Wiener filter gains could be determined. Computation of global and local image statistics, as well as Wiener filtering, is well known in the art and will not be discussed further.

15   <u>Performance</u>

The gradient-corrected linear interpolation method and system were compared to several others types of interpolation techniques. This comparison was performed using a well-known public-domain Kodak image set, simulating a Bayer sampling array (simply by keeping only one of each of the RGB values for

20   each pixel), and then applying the various interpolation algorithms. This subsampling approach is not really representative of digital cameras, which usually employ careful lens design to effectively perform a small amount of low-pass filtering to reduce the aliasing due to the Bayer pattern subsampling. However, since all papers cited above perform just subsampling, with no low-

25   pass filtering, the same was done so that results could be compared. In addition, all interpolation methods were tested with small amounts of Gaussian low-pass filtering before Bayer subsampling, and it was found that the relative performances of the methods are roughly the same, with or without filtering. The gradient-corrected linear interpolation method and system showed an

30   improvement in peak-signal-to-noise ratio (PSNR) over bilinear interpolation techniques. Moreover, except for Freeman's interpolation technique, most

approaches lead to more than a 5 dB improvement. On average, the gradient-corrected linear interpolation method and system outperforms most other interpolation techniques.

5   Besides PSNR measurements, visual quality was verified using one of the images in the Kodak set. The gradient-corrected linear interpolation method and system yielded a visual quality similar to a good nonlinear method (such as discussed in the Kimmel paper (cited above)), but with a much lower complexity. Compared to the interpolation technique discussed in the Pei-Tam paper, the

10   gradient-corrected linear interpolation method and system produced slightly less visible color fringing distortion, as well as a lower PSNR, at a slightly reduced computational complexity.

           The gradient-corrected linear interpolation method and system is nearly

15   optimal in a Wiener sense, and in fact outperforms many more complex nonlinear interpolation techniques. Compared to a recently introduced linear demosaicing technique, the gradient-corrected linear interpolation method and system produce a 0.5 dB improvement in interpolated image quality and a 12% reduction in mean-square error. This comes from the same quality for the green channel

20   and about 0.7 dB improvement for the red and blue channels. In addition, compared to the technique discussed in the Pei-Tam paper, the gradient-corrected linear interpolation method and system also leads to a small reduction (roughly 12%) in computational complexity.

25 **V.   Exemplary Operating Environment**
           The gradient-corrected linear interpolation method and system are designed to operate in a computing environment and on a computing device, such as the processing device 370 contained in the digital camera system 310 or in the computing device 380. The computing environment in which the gradient-

30   corrected linear interpolation method and system operates will now be discussed. The following discussion is intended to provide a brief, general description of a

suitable computing environment in which the gradient-corrected linear interpolation method and system may be implemented.

FIG. 9 illustrates an example of a suitable computing system environment in which the gradient-corrected linear interpolation method and system shown in FIG. 3 may be implemented. The computing system environment 900 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 900 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 900.

The gradient-corrected linear interpolation method and system is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the gradient-corrected linear interpolation method and system include, but are not limited to, personal computers, server computers, hand-held, laptop or mobile computer or communications devices such as cell phones and PDA's, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The gradient-corrected linear interpolation method and system may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types. The gradient-corrected linear interpolation method and system may also be practiced in distributed computing environments where tasks are performed by remote

processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices. With reference to FIG. 9, an exemplary system for implementing the gradient-

5   corrected linear interpolation method and system includes a general-purpose computing device in the form of a computer 910.

Components of the computer 910 may include, but are not limited to, a processing unit 920, a system memory 930, and a system bus 921 that couples

10   various system components including the system memory to the processing unit 920. The system bus 921 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus,

15   Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

The computer 910 typically includes a variety of computer readable media.

20   Computer readable media can be any available media that can be accessed by the computer 910 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile removable and

25   non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data.

Computer storage media includes, but is not limited to, RAM, ROM,

30   EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape,

magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 910. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a

5   modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media.

Note that the term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode

10   information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

15

The system memory 930 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 931 and random access memory (RAM) 932. A basic input/output system 933 (BIOS), containing the basic routines that help to transfer information between elements

20   within the computer 910, such as during start-up, is typically stored in ROM 931. RAM 932 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 920. By way of example, and not limitation, FIG. 9 illustrates operating system 934, application programs 935, other program modules 936, and program data 937.

25

The computer 910 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. 9 illustrates a hard disk drive 941 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 951 that reads from or writes

30   to a removable, nonvolatile magnetic disk 952, and an optical disk drive 955 that

reads from or writes to a removable, nonvolatile optical disk 956 such as a CD
ROM or other optical media.

5   Other removable/non-removable, volatile/nonvolatile computer storage
media that can be used in the exemplary operating environment include, but are
not limited to, magnetic tape cassettes, flash memory cards, digital versatile
disks, digital video tape, solid state RAM, solid state ROM, and the like. The
hard disk drive 941 is typically connected to the system bus 921 through a non-
removable memory interface such as interface 940, and magnetic disk drive 951
10   and optical disk drive 955 are typically connected to the system bus 921 by a
removable memory interface, such as interface 950.

The drives and their associated computer storage media discussed above
and illustrated in FIG. 9, provide storage of computer readable instructions, data
15   structures, program modules and other data for the computer 910. In FIG. 9, for
example, hard disk drive 941 is illustrated as storing operating system 944,
application programs 945, other program modules 946, and program data 947.
Note that these components can either be the same as or different from
operating system 934, application programs 935, other program modules 936,
20   and program data 937. Operating system 944, application programs 945, other
program modules 946, and program data 947 are given different numbers here to
illustrate that, at a minimum, they are different copies. A user may enter
commands and information into the computer 910 through input devices such as
a keyboard 962 and pointing device 961, commonly referred to as a mouse,
25   trackball or touch pad.

Other input devices (not shown) may include a microphone, joystick, game
pad, satellite dish, scanner, radio receiver, or a television or broadcast video
receiver, or the like. These and other input devices are often connected to the
30   processing unit 920 through a user input interface 960 that is coupled to the
system bus 921, but may be connected by other interface and bus structures,

such as, for example, a parallel port, game port or a universal serial bus (USB). A monitor 991 or other type of display device is also connected to the system bus 921 via an interface, such as a video interface 990. In addition to the monitor, computers may also include other peripheral output devices such as speakers

5   997 and printer 996, which may be connected through an output peripheral interface 995.

The computer 910 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 980.

10   The remote computer 980 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 910, although only a memory storage device 981 has been illustrated in FIG. 9. The logical connections depicted in FIG. 9 include a local area network (LAN) 971

15   and a wide area network (WAN) 973, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 910 is

20   connected to the LAN 971 through a network interface or adapter 970. When used in a WAN networking environment, the computer 910 typically includes a modem 972 or other means for establishing communications over the WAN 973, such as the Internet. The modem 972, which may be internal or external, may be connected to the system bus 921 via the user input interface 960, or other

25   appropriate mechanism. In a networked environment, program modules depicted relative to the computer 910, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 9 illustrates remote application programs 985 as residing on memory device 981. It will be appreciated that the network connections shown are exemplary and

30   other means of establishing a communications link between the computers may be used.

VI.   System Components

The gradient-corrected linear interpolation system 300 shown in FIGS. 3A and 3B includes a number of program modules that allow the system 300 to

5   produce a high-quality interpolation of missing color at a pixel in a quick, simple and efficient manner. FIG. 10 is a block diagram illustrating the details of the gradient-corrected linear interpolation system 300 shown in FIG. 3. The program modules contained in the system 300 now will be discussed.

10   In general, referring to FIG. 10, the gradient-corrected linear interpolation system 300 inputs color information including the raw, mosaic-patterned pixel data 360 produced by the image sensor array 315, processes the information, and outputs one or more missing color values for each pixel 320. More specifically, the gradient-corrected linear interpolation system 300 includes a

15   current pixel selector 1000 that selects a current pixel having a current color from the image sensor array that represents a color image. An interpolation module 1010 computes a first interpolation of color value that is missing and desired at the current pixel. The interpolation uses pixels having the desired color with a region of support.

20

The gradient-corrected linear interpolation system 300 also includes a correction term module 1020 that computes a correction term for the interpolation. Preferably, the correction term is computed from a gradient of pixels within a region of support having the color of the current pixel. Once the

25   interpolation and correction term are computed, a linear combination module 1030 linearly combines the interpolation and correction term to obtain the desired (or missing) color value at the current pixel 320. Preferably, the interpolation and correction term are added together to obtain a correct interpolation that estimates the desired (or missing) color value at the current pixel 320.

30

FIG. 11 is a block diagram illustrating the details of the correction term computation module 1020 shown in FIG. 10. Generally, an interpolation of the desired color at the current pixel 1100 is input into the correction term computation module 1020, a correction term is computed, and an adjust gradient correction is output 1110.

5

In particular, the correction term computation module 1020 includes a gradient-correction selector 1120 that selects the amount of correction that will be applied. A region of support module 1130 defines a region of support around the current pixel. Preferably, the region of support is a 5x5 pixel array centered at the current pixel.

10

The correction term computation module 1020 also includes a gradient correction module 1140 that computes a gradient correction using the current pixel. In addition, the module 1140 uses pixels within the region of support that are of the current color, in other words, the same color as the current pixel. A gradient-correction gain application module 1150 applies the gain to the computed gradient correction such that an adjusted gradient correction is output 1110.

15

20

The foregoing description of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description of the invention, but rather by the claims appended hereto.

25